

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03



Test Plan for Applications joining State of Georgia Portal

(for applications developed by an Agency or Third-party vendor)

State of Georgia Technology Authority
100 Peachtree Street Suite 100
Atlanta, Georgia 30303-3404

Version 1.8
March 10, 2003

Revision History

Date	Version	Author
March 10, 2003	1.8	GaNet Quality Management Unit – Test Team

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Background	4
1.3	Scope/Phases	4
1.3.1	Test Stages outside of scope	4
1.3.2	Test Stages within scope	5
1.4	Definitions, Acronyms and Abbreviations	5
1.5	Test Approach	5
1.5.1	Test Goals/Objectives	5
1.5.2	Test Risks	5
1.5.3	Assumptions	6
2.	Requirements for Test	6
2.1	Project Documentation	6
2.2	Requirements	7
3.	Test Strategy	8
3.1	Test Process Analysis	8
3.2	Entrance and Exit Criteria for AUT Test Process	9
3.2.1	Entrance Criteria for Portal System Test	9
3.2.2	Evaluation of Exit Criteria from System Test	9
3.2.3	Exit Criteria from System Test	9
3.3	Test Types	9
3.3.1	Function Testing	10
3.3.2	User Interface Testing	10
3.3.3	External Interface Testing	11
3.3.4	Database Integrity Testing	11
3.3.5	Load, Stress, and Performance Testing	12
3.3.6	Security and Access Control Testing	12
3.3.7	Configuration Testing	13
3.3.8	Regression Testing	13
3.4	Test Tools	13
4.	Defect Tracking	14
4.1	Defect Severity and Priority	14
4.2	Defect Tracking Process	14
4.3	Reports	15
4.4	Metrics	15
5.	Version Control	16
6.	Resources	16
6.1	Roles	16
6.2	System Resources	19
7.	Test Environment Setup	19
8.	Test Schedule	21

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

8.1	Project Milestones	21
9.	Approvals	21
10.	APPENDIX A – Development and Testing Tools	22
11.	APPENDIX B – Test Schedule	22
12.	APPENDIX C – Georgia.gov Portal Environment Application Migration Overview	23

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Portal Applications Test Plan

1. Introduction

1.1 Purpose

This Test Plan outlines and defines the strategy and approach to be taken to perform testing on Web Applications, developed outside of GTA, that are to join the georgia.gov Portal site. It is intended to be instructional, explaining the process used for Portal Acceptance Testing; it is also a template, and may be adapted for each Web Application that is joining the Portal. Instructions throughout this document are boxed and in italics. The GTA Test Unit will work with the Agency owner and/or Project Manager of the application to be tested to prepare this document.

Fill in the blanks of the following paragraphs, and hereafter refer to the 'Agency' and the 'Application Under Test'.

The purpose of this Test Plan is to insure that the application, _____ (hereafter referred to as the "Application Under Test" or "AUT"), to be placed on the georgia.gov portal site, meets the necessary requirements as defined in:

- Agency requirements for this application (defined by Use Cases, Supplemental Specifications, and/or other means)
- GTA Content Management Specifications
- GTA/EPIA portal standards referenced in the Joining georgia.gov Guide

The Georgia Technology Authority (GTA) and the business owner agency/entity, _____ (hereafter referred to as the "Agency"), will be the primary users of this Test Plan. This Test Plan will outline the scope, test requirements, strategy, approach, responsibilities, and will define the tests to be used for testing this Web Application. End-to-end testing will be conducted, with primary focus on the functional testing of the critical and high-risk functions of an application with the Portal, via the web user interface. This process will be referred to as Portal Acceptance Testing.

The GTA GaNet Quality Management Test Unit will be responsible for conducting or insuring that the Portal Acceptance Tests, outlined below, are conducted. A Project Team will be identified to insure adequate representation from all vested entities (e.g. Agency, Developers, Test Unit, IRM, Security, etc.). The QM Test Team will coordinate communication among this Project Team and arrange for sign-off for moving the application to the Portal.

1.2 Background

This Test Plan applies to the State of Georgia portal web site, georgia.gov, hereafter referred to as The Portal. The testing process outlined in this document is based on the Rational Approach to software development and testing. The Rational Approach involves a coherent, well thought-out methodology and a set of powerful tools. This methodology for software development, Rational Unified Process or RUP, is designed to be adapted by an organization; Georgia's adaptation is referred to as the Georgia Unified Process, GUP. This Test Plan is based on the Test Plan template in the RUP and is stored in the GUP repository. Where possible, Rational tools will be employed. Test procedures will be identified and tracked using Rational Test Manager.

1.3 Scope/Phases

Target of test is:

End to End testing of the AUT with the State of Georgia portal, via the web user interface

1.3.1 Test Stages outside of scope

The Software Development Group (Agency or Out-sourced Vendor) that develops the application and/or The Agency is expected to conduct the following tests for the standalone application, prior to submitting a request to join The Portal. These tests are outside of the scope of this Test Plan. (NOTE: for software developed by GTA, the following items will be in scope.)

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

- Unit Testing will verify that a single component functions as it should with respect to the requirements it implements; the Software Developers should conduct this.
- Integration Testing will verify that the software components that comprise the application work together, and should be conducted by the Software Developers.
- The Developers and Agency will conduct System Testing for the Web Application in a Test environment and then in the Production environment. All system testing should be conducted iteratively in phases throughout the development cycle.
- Acceptance Testing should be conducted by the Agency in order to accept the AUT as meeting the stated requirements, including readiness for placing on The Portal.

1.3.2 Test Stages within scope

The Project Team will conduct Acceptance Testing for the functioning of the AUT within The Portal, to meet the Target of Test stated above. Time and resource restraints will not permit a repeat of all the system tests that are expected to be performed by the developers and Agency (see above) throughout the development cycle of the system. The following tests will be conducted and are explained in-depth in Section 3:

- Function Test - based on Agency Requirements (this will focus on the critical and high-risk functions of the system)
- User Interface Test - based on the EPIA Style Guide
- External Interface Test – based on External Interface Link Standards
- Database Integrity Test – based on database requirements and interactions as specified by competent DBA resource.
- Load, Stress, Performance Tests – based on Load, Stress, Performance Standards
- Security and Access Control Test – based on Enterprise Information Security Policies

1.4 Definitions, Acronyms and Abbreviations

Refer to the GTA Glossary (on the EPIA Repository site [Document Repository](#))

AUT – Application Under Test

SDG – Software Development Group (may be within GTA, an Agency, or an Outsourced Vendor)

GUP – Georgia Unified Process (adaptation of Rational Unified Process)

Project Team – will include all vested entities (e.g. Agency, Developers, Test Unit, IRM, Security, etc.). The QM Test Team will coordinate communication among the Project Team

ART – Agency Request for Technology that is submitted to GTA GaNet, through a GTA Account Manager

SRS – Software Requirements Specifications – an artifact of RUP, a complete description of the requirements for a project, includes Use Cases, etc.

Jgg – Joining georgia.gov Guide

EPIA – Enterprise Portal Interoperability Architecture

1.5 Test Approach

Testing should only be executed using known, controlled databases, and in secured environments. Typically, to develop the AUT test approach, the GTA/Agency Project Team will review system requirements (in the form of Use Cases and Supplemental Specs, etc.) and other documentation.

See Appendix C – Georgia.gov Portal Environment Application Migration Overview.

1.5.1 Test Goals/Objectives

The primary focus will be end to end testing of an application with the Portal, via the web user interface. This process will be referred to as Portal Acceptance Testing. The goals are to determine:

- Does the application operate as planned on the Portal?
- Does it comply with the Portal look and feel?
- Does it work without damaging any other aspects of the Portal?

1.5.2 Test Risks

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Remove or list any additional risks:

- Test Team may be new to Rational Unified Process and Tools
- Offsite development may contribute to communication risks in defect resolution, especially without a common defect-tracking tool
- Cross browser functionality verification/validation is very time consuming
- If the Test environment is not a mirror of Production environment, this will introduce variables in test results

1.5.3 Assumptions

Remove or list any additional Assumptions:

- Project Team Composition - A Project Team will be identified to insure adequate representation from all vested entities (e.g. Agency, Developers, Test Unit, IRM, Security, etc.). The QM Test Team will coordinate communication among this Project Team, provide feedback and suggest improvements and arrange for sign-off for moving the application to the Portal.
- GTA Role – In addition to ensuring a thorough portal acceptance test, according to this Test Plan, GTA will mentor the Agency, as necessary, on use of GUP and GTA standards.
- Test Performance - The SDG will have conducted a thorough test and the AUT will be fully functional in a standalone environment (not on the portal).
- Early involvement – When possible, the GTA QM Test Unit should be involved with the AUT in the early stages, and participate in review of Requirements documents, Functional Test documents and early system tests.
- Test Limitations – Given resource limitations and the limitless number of test paths and possible input values, the test effort will focus on the most critical and high-risk functions of the system.
- Project Schedule – It is assumed that both agencies will establish a schedule and adhere to the schedule that is agreed upon.
- Build Versioning – All builds of the AUT submitted for testing must have version control implemented to allow for Build Found, Resolved, and Tested during the defect tracking process.
- Hardware/Software - All necessary hardware and software will be available when appropriate. Specific platform/browser configurations are outlined in detail in this document (see Resources, Section 6).
- Process - Testing for this AUT will follow the process outlined in this Test Plan, although different testing tools may be used.
- Acceptance Criteria – Acceptance Criteria will be established by identifying objective completion criteria for determining acceptability of the AUT for joining The Portal (see Section 3, Test Types and Exit Criteria; Section 4, Severity and Priority).

2. Requirements for Test

2.1 Project Documentation

The table below identifies the documentation that may be used for gathering requirements and developing the Test Plan for the AUT.

List those that are to be used; check those that are available and received:

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Document/Deliverable Type	Deliverable Name	Created?	Received?	Author or Resource
Software Requirements Specification	<i>Examples – Use Cases; Supplementary Specifications (those requirements not readily captured in Use Cases); Mock-ups of Look and Feel; Wire frames</i>			
Password Security Document	<i>User Types and Security Permissions</i>			
Project Plan that includes dates anticipated for Testing	<i>Examples–ART/Traffic Proposal; MS Project Workspan Statement of Work (SOW)</i>			
Screen Design Document				
Business Model or Application Flow?	<i>Ex - User Interface Prototype to include wire frames and screen mock-ups to illustrate navigation and look and feel of the application; Web Application Flow Charts that describe user navigation</i>			
Business Functions and Rules	<i>Ex. - Use Cases</i>			
Software Architecture Documents	<i>Ex. – Architectural Views; Data Dictionary</i>			
Interface Control Documents (for each external interface?)				
Data Conversion Mapping				
Standards	<i>See Appendix for Portal Standards</i>			

2.2 Requirements

The Requirements for Test further identifies those items—Use Cases, functional requirements, and non-functional requirements—that have been identified as targets for testing (for applications developed outside of GTA, critical and high level Use Cases will be required). This list is taken from the Document/ Deliverable Table above. For applications developed outside of GTA, critical and high level Use Cases will be required.

Here you may elaborate on the documents to be used, listing all the Use Cases that are provided, Supplemental Specifications, etc., as necessary. The Agency is expected to provide Use Cases and/or documentation for the critical and high-risk functions of the system, and to prioritize Requirements.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

3. Test Strategy

3.1 Test Process Analysis

The Test project will use GTA's standard test process outlined in this document that adopts the desired areas of the Rational Unified Process (RUP) into the Georgia Unified Process. The following steps are typically necessary for testing an application that is joining the Portal.

Make changes to this process as necessary for the current AUT.

1. Prior to Portal Acceptance Testing, the Agency (Business Owner of the Application to be tested) and the Software Development Group (in or out of GTA) must have successfully conducted Unit, Integration, System, and Acceptance Testing for the standalone application.
2. The Agency and Software Development Group will have complied with all applicable Portal Standards (see list attached) where possible. Any exceptions will be discussed and approved in advance by the Project Team.
3. The QM Test Unit will be involved early in the development process so that -- requirements will be familiar to the Testers, test preparation can occur in advance, and some of the tests below can be conducted iteratively.
4. A Project Team will be identified to insure adequate representation from all vested entities (e.g. Agency, Developers, Traffic Project Manager, QM Test Unit, IRM, Security, etc.). The QM Test Unit will coordinate communication among this Project Team and arrange for sign-off for moving the application to the Portal.
5. The Agency Business Owner (if development occurs outside of GTA) or the GTA Software Development Lead (if development occurs within GTA) will submit an application to be tested to the Director of the GaNet Quality Management Unit. It is expected that testing will be scheduled ahead of time and that adequate time will be allowed for thorough testing as determined by the QM Test Unit.
6. The Agency Business Owner will provide Requirements documents, Functional Test documents and Test Data and some method to allow full verification and validation of test data to be used for testing.
7. The Developers must provide a Technical Design including application files, calls, and locations to the GTA Configuration Manager.
8. The GTA Configuration Manager will set up the Test Environment (hardware and software). This is expected to mirror the Production Environment. The test process will follow established Version Control strategy and Iteration/Build nomenclature found in the GTA QM Configuration Management Plan.
9. The QM Test Unit will coordinate Planning Tests:
 - Identify Requirements for Test/Prioritize Requirements (opt: Baseline and Tag in ReqPro)
 - Identify Resources (equipment and staff)
 - Create Test Schedule
10. The QM Test Unit will coordinate Designing Tests:
 - Generate Test Model (collection of test cases, procedures, scripts, expected results)
 - Determine Severity Criteria on which to base successful test outcomes
11. The QM Test Unit will coordinate Implementing Tests:
 - Prepare Test Scripts (if tests are to be automated)
 - Establish Test Data
 - Determine Defect Tracking Procedure
12. The QM Test Unit will coordinate Executing the following Portal Acceptance Tests based on Joining the Portal Standards (listed in table below); these will be conducted iteratively if multiple builds are necessary:

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

- Function Test - based on Agency Requirements (this will focus on the critical and high-risk functions of the system)
- User Interface Test - based on the EPIA Style Guide
- External Interface Test – based on External Interface Link Standards
- Database Integrity Test – based on Database Integrity Standards
- Load, Stress, Performance Tests – based on Load, Stress, Performance Standards
- Security and Access Control Test – based on Enterprise Information Security Policies

13. The QM Test Unit will coordinate Evaluating Tests:

- Analyze Defects resolution; Evaluate Results using Success criteria; and Produce Report
- Determine, with Project Team, when application is ready for placement on the Portal

14. The GTA Configuration Manager will move the Application to the Production Environment.

3.2 Entrance and Exit Criteria for AUT Test Process

In order to verify that the AUT test for joining the Portal has been completed, certain criteria must be met. Below is a description of the criteria that must be satisfied in order to enter and exit from System Test.

3.2.1 Entrance Criteria for Portal System Test

- Test Planning is in place
- Agency has provided Requirements Specifications
- Agency has provided list of User Types and Permission for Security testing
- All Unit and string/integration tests conducted by the Vendor/Agency have been executed and passed. A results summary from the vendor/software developer will be necessary for GTA to verify completion.
- Version control is in place for AUT.

3.2.2 Evaluation of Exit Criteria from System Test

Evaluation of the Exit Criteria from System Test is used to determine the “Go” or “No Go” decision for deployment of AUT on the Portal.

- A determination must be made on whether the mandatory business requirements have been met:
 - Have all high priority test conditions been satisfied?
 - Of those test conditions not satisfied, how many will be needed Day 1 after deployment?
 - Of those test conditions not satisfied and needed Day 1, is there a viable workaround?
- Once any un-met requirements have been determined, goals can be set for proper testing within the time frame remaining in the project schedule, in order to achieve a “Go” decision for deployment.

3.2.3 Exit Criteria from System Test

The following is a list of criteria that must be met in order to sign-off on System Test:

- All Critical and High severity (see Section 4, Severity) defects have been resolved.
- All other defects have been reviewed by the Test Manager and Business Manager (Agency owner or project manager proxy) and approved to go into production.

3.3 Test Types

Following are the types of tests that will be conducted. See ‘6, Resources, Roles’ for who will perform each test.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

3.3.1 Function Testing

Functional Testing verifies proper data acceptance, processing, retrieval, search findings, and the appropriate implementation of the requirements, using valid and invalid data for the application under test (AUT). For applications developed outside of GTA, functional tests will be conducted only for critical and high-risk functions, and Use-Case flow (it is expected that Agencies will have previously conducted a thorough Functional Test).

Test Objective:	Ensure proper application functionality, including navigation, data entry, processing, retrieval, update, and search.
Technique:	Execute each Use Case for critical and high risk functions, and Use-Case flow, using valid and invalid data, to verify the following: <ul style="list-style-type: none"> • The expected results occur when valid data is used in all test cases. • The appropriate error or warning messages are displayed when invalid data is used. • The appropriate information is retrieved from this Web Application. • The appropriate information is updated to this Web Application. • Critical business rules are properly applied.
Test Resources:	Use Cases, Agency Business Rules, Supplemental Specifications
Completion Criteria:	<ul style="list-style-type: none"> • All planned tests have been executed. • All identified defects have been addressed to the satisfaction of GTA and Business Owner of Web Application.
Special Considerations:	Availability of test data.

3.3.2 User Interface Testing

The goal of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the application. In addition, UI testing ensures that the objects within the UI function as expected and conform to Portal or industry standards.

Test Objective:	Verify the following: <ul style="list-style-type: none"> • Navigation through the AUT properly reflects business functions and requirements. For the web user interface this includes window-to-window, field-to-field, and use of access methods (tab keys, mouse movements). • Window objects and characteristics, such as menus, size, position, state, and focus conform to Mockups or other descriptions for the look & feel, including Portal Style Guide. • Proof for spelling and grammar.
Technique:	Create or modify tests for each window or menu to verify proper navigation and object states/values for each application window, prompt, and object.
Test Resources:	Portal Style Guide, GTA Standard for Error Messages, Agency Requirements, Agency Business Functions, Mockups
Completion Criteria:	Each window or prompt is successfully verified to remain consistent with mockups and/or wire frames and Portal Style Guide.
Special Considerations:	

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

3.3.3 External Interface Testing

External Interface (EI) testing verifies the AUT's interaction with each external interface at the portal to that system. The goal of EI testing is to ensure that the External Interface provides the user with the appropriate access and data passage through the functions of the AUT. In addition, EI testing ensures that the objects within the EI function as expected and conform to corporate or industry standards.

Test Objective:	Verify the following: <ul style="list-style-type: none"> • Navigation through each external interface functions properly and reflects business functions and requirements stated in the Use Cases. • Data is passed between systems properly.
Technique:	Create or modify tests for each interface to verify conversations, proper navigation and object states for each application interface.
Test Resources:	Agency Requirements
Completion Criteria:	Each testable interface successfully verified to remain consistent with benchmark version or within acceptable standard.
Special Considerations:	<ul style="list-style-type: none"> • Not all properties for third party interfaces can be accessed. • Will require an "expert" for each interface. • Will need full documentation for each interface, and an Interface Control Document should be required for each interface. (The system analysts in concert with the development team and the external systems teams will author these documents.). • Test team confirms data at the interface. Any testing of the other systems interfaced is the responsibility of the appropriate GTA/Agency technical and business team that owns the system. • Upgrades and enhancements to 3rd party systems during the development and testing lifecycle need to be communicated to the Test Team Lead.

3.3.4 Database Integrity Testing

The Agency databases and the database processes will be tested as a subsystem within the GTA portal.

Test Objective:	Ensure Agency Database access methods and processes function properly and without data corruption.
Technique:	<ul style="list-style-type: none"> • Invoke each database access method and process, seeding each with valid and invalid data or requests for data. • Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved for the correct reasons.
Completion Criteria:	All database access methods and processes function as designed and without any data corruption.
Special Considerations:	<ul style="list-style-type: none"> • Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases. • Processes should be invoked manually. • Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events. • DBA will need to supply the full requirements for the data relationships and mapping on which to base test cases. • The production environment needs to mirror the test environment to every extent possible.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

3.3.5 Load, Stress, and Performance Testing

Load, Stress, and Performance testing is designed to examine whether the system functions under real-world activity levels. This verifies whether the system can handle projected user-volumes and processing requirements.

Test Objective:	The objective of performance testing is to demonstrate that a system functions in accordance with its performance requirement specifications regarding acceptable response times, while processing the required transaction volumes on a production size database. During performance testing, production loads are used to predict behaviour and a controlled and measured load is used to measure response time. The analysis of performance test results helps support performance tuning. Stress testing involves the process of running the client machines in high-stress scenarios to see when and if they break.
Technique:	<ul style="list-style-type: none"> For the AUT, record a baseline of performance test scripts developed against the initial system. Verify that the system meets the performance requirements. Test first in the Test environment; test in the Production environment.
Test Resources:	GTA Standards; Agency provides # of expected concurrent users
Completion Criteria:	<ul style="list-style-type: none"> Systems meet or exceed performance requirements.
Special Considerations:	<ul style="list-style-type: none"> There will be baseline standards from the existing systems provided by the GTA IRM, Systems Support. These types of tests require specialized software.

3.3.6 Security and Access Control Testing

These tests will be based on the EPIA standards that address the overall architecture and strategy for security and access control testing.

Security and Access Control Testing focus on two key areas of security:

- Application security, including access to the Data or Business Functions, and
- System Security, including logging into / remote access to the system.

Test Objective:	<p>Application Security: Verify that user can access only those functions / data for which their user type is provided permissions.</p> <p>System Security: Verify that only those users with access to the system and application(s) are permitted to access them.</p>
Technique:	<ul style="list-style-type: none"> Function/Data Security: Identify and list each user type and the functions/data each type has permissions for. Create tests for each user type and verify each permission by creating transactions specific to each user type. Modify user type and re-run tests for same users. In each case verify those additional functions / data are correctly available or denied. System Access (see special considerations below)
Completion Criteria:	<ul style="list-style-type: none"> For each known user type the appropriate function / data are available and all transactions function as expected and run in prior Application Function tests.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Special Considerations:	<ul style="list-style-type: none"> Access to the system must be reviewed / discussed with the appropriate network or systems administrator. This testing may not be required as it may be a function of network or systems administration.
-------------------------	---

3.3.7 Configuration Testing

Configuration testing verifies the operation of the application on the required hardware and software configuration, focusing particularly on a combination of different browser configurations and different client-side operating systems..

Test Objective:	Verify that the application functions properly on the required client hardware and software configurations.
Technique:	Execute all the test scripts on different browser configurations and different client operating systems.
Test Resources:	Browser configurations in GTA Test Plan
Completion Criteria:	Same as for Function Testing
Special Considerations:	See Section 6, System Resources for configurations that will be tested.

3.3.8 Regression Testing

Regression Testing is done at the end of each iteration to determine if any fixes have introduced other errors. This will be done if multiple builds are necessary for Portal testing.

Test Objective:	Verify that all of the application functions properly after code changes in new builds.
Technique:	Run the test cases of the previous iteration. Automate when possible. There will not be a formal Regression Testing stage; instead, regression testing will be conducted as needed.
Test Resources:	All sources listed above
Completion Criteria:	All planned tests have been executed. All identified defects have been addressed.
Special Considerations:	

3.4 Test Tools

The Test Team for this project will use the test tools listed in Appendix A.

Test Scripts will be developed from Test Cases if needed. For acceptance testing, it is usually not feasible to automate tests, unless they were automated in previous iterations. Test results will be tracked in Rational Test Manager, whether automated or manual tests are performed.

If software problems are detected, the team will report to system developers through the approved tracking system. The Rational project database supports the repository for system requirements, test requirements, and related software problem reports.

Rational RequisitePro will serve as the requirements management tool; this provides a link from the requirement to the test case through integration with Rational TestManager.

List the tools to be used in Appendix A.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

4. Defect Tracking

4.1 Defect Severity and Priority

“Severity” refers to specific program or system behavior as a result of defects.

Severity Level	Type	Description
1	Critical	System crash, run-time error, issue blocks use of program
2	High	Loss of functionality based on functional requirements
3	Medium	Loss of functionality not traceable to functional requirements
4	Low	Cosmetic or other non-functional defect

“Priority” characterizes the business impact of the defect and attaches varying degrees of importance to repair of the defect. Some 1-Critical defects have little impact on the project goals because users may, in practice, never experience them. Some 3-Medium defects can become high priority fixes because the functionality is highly desirable to the business agency or users. If necessary, the Test Team will prioritize defects in addition to assigning a “severity” level.

4.2 Defect Tracking Process

To track and report defects, the defect workflow process below will be implemented to record and report defects and change requests.

Defect Reporting Process

Defect correction is the responsibility of system developers; defect detection is the responsibility of the Test Team. The Test Team Lead will manage the defect tracking system, and will report defects to appropriate people. Suggested steps with the associated status in the defect reporting workflow process are as follows:

- Submitted** -When a defect is found, the tester notifies system developers by entering the defect into ClearQuest, selects the severity of the defect, and sets the status to “Submitted”. Testers will add any attachments, such as a screen print, relevant to the defect.
- Approved/Rejected/Postponed**
 - **Approved** is assigned if it is a valid defect.
 - **Rejected** is assigned if the defect report is based on user error or if a duplicate or is as designed. (The reason for “rejecting” the defect needs to be documented.)
 - **Postponed** is assigned if no one is available to work on it or for other reasons work must be delayed, where something is on hold.
- Assigned** - If the status is determined to be “Approved,” the development manager (or other designated person) assigns the defect to the responsible person (developer) and sets the status to “Assigned.”
- Opened** – a developer will change to an open status when the assignment is received (or work begins?).
- Unit Tested** – Developers will test. After the defect has been fixed, the developer documents the fix in the defect-tracking tool and sets the status to “Unit Tested.” [(At the same time, the developer reassigns the defect to the original submitter. The developer must indicate a build number in which the defect is or is to be repaired. The system developers will correct the problem in their facility and implement the operational environment after the software has been baselined. Notes that detail the defects corrected will accompany each release/build)
- Built** – The Configuration Manager has placed the defect/change on (test) server for testing.
- Re-work Required** – if a defect is found, the defect/change request will be sent to developer for more work and status will be changed to “Re-work Required”.
- QM Tested** – defect/change has successfully been tested

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

9. **UAT Tested** – the agency owner has successfully tested the defect/change.

10. **Deploy Production Tested** – QM has tested deploy to production.

11. **Closed** - If the defect has been corrected with the fix, the tester sets the status to “Closed.”

4.3 Reports

Defect Reports – will report program errors or omissions that cause the program to fail to meet functional specifications, or fail to provide for requirements.

Change Requests – will report user or business owner requests for program changes to allow for changed or re-interpreted user requirements.

The development manager, test manager, and business owner (or project manager proxy) can convert a defect report into a change request by recognizing that a defect reported identifies an unmet business need or an incidence where requirements were incomplete or ambiguous, requiring rework of requirements and subsequent program changes. Many defects marked “HOLD” will typically be converted into Change Requests for subsequent builds or releases.

4.4 Metrics

This table shows the test metrics that can be collected and reported.

Testing Metrics

Metric Name	Description
Test Procedure Execution Status	Number of executed test procedures versus total number of test procedures. This metric will indicate the extent of the testing effort still outstanding.
Error Discovery Rate	Number of total defects found versus number of test procedures executed. It is used to analyze and support an intelligent product release decision.
Problem Reports by Severity	Open defects, sorted by severity, with a defect number and one line description for each defect. Each defect priority should also be shown.
Problem Reports by Priority	Number of software problems reported, listed by priority.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

5. Version Control

Software Configuration Management establishes and maintains the integrity of software products throughout software project's life cycle and ensures all parties involved are kept informed of each product's status as it evolves from requirements into a physical entity. It may be necessary to rebuild an application if defects are found; if so, the current baseline version must not be deleted or lost as new versions are created. All versions from the first to the last must be kept and archived. A Rational project will be set up for this purpose for this AUT.

Builds will be made available in the test environment based on the development and test schedules and it is the responsibility of the vendor/developer to provide the configuration manager with the necessary documentation to place all necessary software components on the appropriate servers. The Test Environment Manager will work with the vendor/developer and the DBA to assure operation is completed successfully. Each build will have an incremental increase in version number stated and visible to test team for the logging of defects.

To accomplish this, GTA Configuration Management in conjunction with GTA Project Management will baseline all accepted deliverables and major new releases of deliverables. Source code will be stored in Configuration Management System (CVS). A Rational project will be established for each AUT; here artefacts, documentation, etc. used for testing will be placed. These documents must all be in Word or Adobe (.pdf) format. Upon acceptance of the document deliverables, the owner will deliver the document in its original format for baselining in the GTA Configuration Management system.

The vendor will provide release notes with each build stating all version information, functionality included in the build and list of defects fixed from previous build

The following File Naming conventions have been established for all artifacts for the AUT Project,

<AUT><DOCUMENT TYPE>_V<VER (Use the 4 Levels below)>

Level 1 - Major Release (Release 2 vs. Release1)

Level 2 - Minor Release (Very significant, such as at baseline times)

Level 3 - Significant (significant enough to track, but not worthy of a minor release, such as a new requirement added)

Level 4 - Work in Progress (for daily saves; the developers are doing nightly check ins; others should also do daily check ins if changes are made)

EXAMPLE: AUT_ConfigMgtPlan_V1.2.4.7 (Assuming there is no ConfigMgtPlan from Rel1);

otherwise: AUT_ConfigMgtPlan_V2.2.4.7

6. Resources

6.1 Roles

This table shows the staffing assumptions for testing the AUT.

This is a guide; at times, one person may fill more than one role. Add the name, email address, and phone number of the persons who will fill these roles:

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Staff Resources and Responsibilities

Role	Source	Name, email, Phone	Responsibilities/Comments
Test Manager	Provided by GTA GaNet		Manage testing effort <ul style="list-style-type: none"> • Estimate and manage resources • Create and manage schedule for testing effort • Report to stakeholders • Provides technical direction to staff • Evaluate effectiveness of test effort • Manage completion of Planning effort, (i.e. status tracking and deliverable mgt.) and Test Plan • Assure requirements analysis documentation • Coordinate Schedule, including Build Schedule, Dates • Assign Roles • Organize and present Test Evaluation Summary
Lead GTA Tester	Provided by GTA GaNet		Provide second level technical direction. <ul style="list-style-type: none"> • Coordinate testing effort for specific systems • Lead review of all test conditions for completeness and consistency. • Lead review of all scripts for completeness and accuracy • Provide mentoring for client and test team • Track and report Defects • Train Test Team on Defect Tracking • Manager version control plan
Lead Agency Tester	Provided by Agency		Provide second level technical direction. <ul style="list-style-type: none"> • Coordinate testing effort for AUT as needed on Agency side
Test Environment Manager and DBA	Provided by GTA GaNet		Ensure all test environments and assets are managed and maintained. <ul style="list-style-type: none"> • Administer test and data environments. • Coordinate with DBA to maintain and refresh data. • Verify and correct any data problems. • Review and follow all data-related issues through the defect tracking process. Includes correcting input files, tracking transactions, and following up with Test Designers to close issues. • Keep track of changes in scripts and test conditions and verify that input data is consistent with these changes. • Test-bed configuration management (CM). Maintain the entire test-bed/repository (that is, test data, test procedures and scripts, software problem reports) in a CM tool. • Set up necessary access to Tools
Business Manager	Provided by Agency		<ul style="list-style-type: none"> • Make Agency resources available for testing. • Review Test Results with Test Manager to determine completion of test process.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Test Designer	Provided by GTA GaNet		<p>Identify, prioritise, and implement test cases</p> <ul style="list-style-type: none"> • Generate Test Cases, defined as high-level requirements for test • Generate Test Procedures, defined as a set of steps and verification points for executing a Test Case, using TestManager or other appropriate software • Generate Test Scripts if Test Cases are to be automated • Coordinate data selection. • Ensure test conditions are up to date with current designs. • Logically group test conditions into scripts and scripts into cases. • Work with the build teams to resolve any test condition or script issues. • Maintain test condition database. • Work with data environment manager to ensure proper set up for scripts. • Review and follow all test condition-related issues through the Defect tracking process.
Testers	Provided by GTA and Agency		<p>Execute the tests</p> <ul style="list-style-type: none"> • Perform Function, User Interface, Extended Interface, Database Integrity, and Configuration testing. • Execute online steps of each assigned Test Case or Script. • Investigate & document issues/defects with data/application. • Consult experts on each build as needed. • Verify and sign-off system test conditions. • Provide a daily status by test condition/script. • Assess next day/week impacts of issues. • Document change requests
Subject Matter Experts (Functional) for Analysis & Support	Provided by Agency		<ul style="list-style-type: none"> • Analyse issues from testers. • Trouble-shoot data problems. • Provide technical expertise on new functionality. • Track migration of fixes in relation to assigned scripts. • Provide cross-support to developers on the resolution of issues, (i.e. recreation of problem in development environment). • Identify issues with data selection.
Software Development Group	Either in a state Agency, or an out-sourced Vendor		<ul style="list-style-type: none"> • Execute unit, software integration, and portal/application integration testing • Follow defect tracking and resolve issues. • Work with Test Manager and Test Designer to release Builds. • Provide Test Results Summary • Provide Source Code • Purchase Tools if necessary • Provide Release Notes with each Build • 3rd Party Vendors – supply certification of test results
Security Lead	Provided by GTA		GTA Office of Information Security will execute Security and Access Control testing.
IRM Lead and GaNet Lead	Provided by GTA		GTA IRM and GaNet will work together to manage Volume/Load, Stress, Performance, Fail-over, and Recovery testing.
Load Tester	Provided by GTA		Execute load, performance, stress, volume, fail-over and recovery testing.
Agency Tasks			<ul style="list-style-type: none"> • Provide Software Requirements Specifications • Provide list of User Types and Permissions (3.5.6)
Software Configuration Manager			<p>SCM acts as controller and communicator:</p> <ul style="list-style-type: none"> • Establish a software baseline library to provide storage for the work products and to provide for controlled access • Identification of the software work products that need to be controlled • Establishment of product baselines • Definition and implementation of processes to systematically control changes to the product baselines • Establishment of roles of the individuals involved in the SCM process. • Configuration Status Accounting • Configuration Audits

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

6.2 System Resources

The following tables set forth the system resources that are needed for testing the application itself and for testing as the user of the application accessing it via a web browser (or telephone).

Some of the possible Browser Configurations to be tested

Client Test Platform	Browser Configurations
Macintosh OS 9.0	IE 5.0
	IE 5.5
	IE 6.0
	Netscape 6
	Netscape 4.7
MS Windows 98	IE 5.0
	IE 5.5
	IE 6.0
	Netscape 4.8
	Netscape 7
Windows 2000	IE 5.0
	IE 5.5 *
	IE 6.0
	Netscape 4.8
	Netscape 7
Telephone	Landline, Cordless Phone, Cellular Phone
	AOL

NOTE – Testing with all the configurations above will also be done with Cookies turned off and JavaScript disabled.

- IE5.5 is the default browser configuration; others are used only at client request.

7. Test Environment Setup

The test environment should mirror the production environment. This section describes the hardware and software configurations that compose the system test environment. The test team will need the assistance of a Test Environment Manager and DBA to resolve issues and perform administration tasks. The hardware must be sufficient to ensure complete functionality of the software. Also, it should support performance analysis aimed at demonstrating field performance. Information concerning the test environment pertinent to the application, database, application server, and network is provided below.

Fill in Test Environment information.

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

Test Environment System Resources

Resource	Name / Type
Application Server	
Network/Subnet	TBD
Server Name	TBD
Database Name	TBD
Web Server	
Network/Subnet	TBD
Server Name	TBD
Database Name	TBD
Database Server	
Network/Subnet	TBD
Server Name	TBD
Database Name	TBD
Database Server	
Network/Subnet	TBD
Server Name	TBD
Database Name	TBD
Client Test PCs	
Include special configuration requirements	TBD
Test Repository	
Network/Subnet	TBD
Server Name	TBD
Test Development PCs	TBD

Test Plan for Web Applications to be placed on the State of Georgia Portal	Version: 1.8
Whatever Web Application	Date: 3-10-03

8. Test Schedule

The Test Schedule will be based on the AUT Project Plan and attached to this document as Appendix B. It will include Milestones and deliverables noted in the Test Process Analysis (3).

8.1 Project Milestones

The overall milestones of the Test Schedule may be presented here.

Fill in this table for an overview of test activities.

Project Milestones

	Milestone Task	Start Date	End Date
Iteration #1			
	Plan Test		
	Design Test		
	Implement Test		
	Execute Test		
	Evaluate Test		

9. Approvals

This section should list the individual responsible for reviewing and approving the Test Plan from each entity assigned tasks in Roles, Section 6. All individuals who must support the testing effort should review this plan.

Fill in names and route for signatures.

Agency/Unit	Name	Signature	Work #	Email address	Date
1.					
2.					
3.					
4.					
5.					

Test Plan	Version: 1.8
Whatever Web Application	Date: 03/7/03

10. APPENDIX A – Development and Testing Tools

The following tools are the GTA standards. When possible, they will be used for software development and testing.

Fill in this table with tools that will be used for this project.

Activity/Task	Tool	Vendor/In-house	Version
Business Modelling	Rational Rose	Rational	2002.05.20
Requirements Management	RequisitePro	Rational	2002.05.20
Test Management	TestManager*	Rational	2002.05.20
Defect Tracking	TBD	Rational	2002.05.20
Test Metrics	TestManager ClearQuest MS Excel MS Access	Rational Microsoft	2002.05.20
Automated Testing Tool		Rational Robot, RobotJ	2002.05.20
Manual Testing (to be used when absolutely necessary)	Test Manager (to record results)	Rational	2002.05.20
Performance Testing Tool			
Test Coverage Monitor or Profiler	TestManager	Rational	2002.05.20
Agency tools	<i>Ex -MS Access; DB2 tools</i>	<i>Ex -Microsoft</i>	
*Test procedures will be identified and tacked using the Rational Test Mgr. This approach will allow for easy management of test progress status. Once a test is performed, the test procedure status is revised within Test Manager to reflect actual test results, such as pass/fail.			

APPENDIX B – Test Schedule

Refer to Section 3.2.1 Test Process to create a Test Plan Schedule of test milestones for AUT. Place the Test Plan Schedule with dates here.

Test Plan	Version: 1.8
Whatever Web Application	Date: 03/7/03

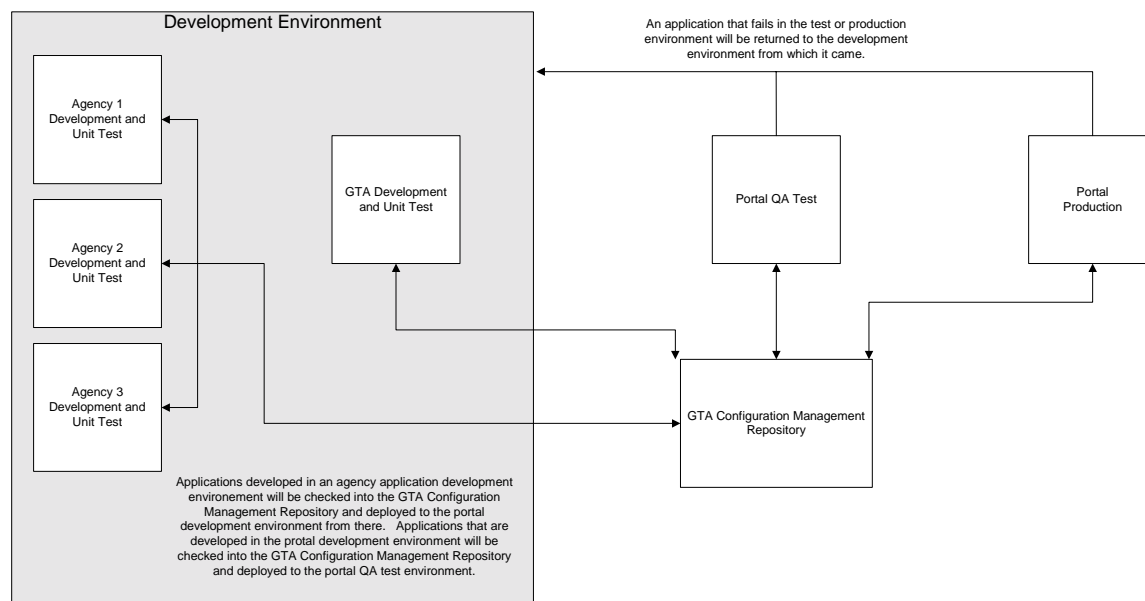
APPENDIX C – Georgia.gov Portal Environment Application Migration Overview

georgia.gov Portal Environment Application Migration Overview

This document provides an overview of the georgia.gov portal environment. The purpose is to show agencies how applications will be promoted from development to production in the portal environment.

Application development can occur in either an agency development environment or, if the agency does not have an application development environment, in the GTA development environment.

georgia.gov Portal Environment Application Migration Overview



Prior to moving an application to the portal environment, the agency must provide deliverables such as a unit test plan showing the results of unit testing, a formal test plan, and complete deployment instructions.

Promotion to GTA Development Environment

An application developed on an agency development environment will enter the portal using the following process:

1. The Agency Application code will be placed in GTA's configuration management repository.
2. It will then be deployed to the GTA development environment using the agency's deployment instructions.
3. While in the GTA Development environment, the agency will conduct system(Unit???) testing to insure the application will function as a complete unit in the portal environment. (Module Testing??)
4. If the application fails testing in the GTA development environment it will be returned to the Agency Development Environment from which it came for further work.

Test Plan	Version: 1.8
Whatever Web Application	Date: 03/7/03

Promotion to GTA Test Environment

1. Upon successful completion of the portal acceptance test and agreement to deploy, the application will be deployed from the configuration management repository to the GTA Portal Test environment using the agency's deployment instructions.
2. In this environment the agency will perform functional testing to insure that the application meets the functional requirements of the user in the portal environment.
3. The agency will also conduct final agency acceptance testing in this environment prior to deployment to production.
4. GTA will conduct portal QA and performance testing to insure that the application meets all portal requirements.
5. If the application fails testing in the GTA Portal Test environment it will be returned to the development environment from which it came for further work.

Promotion to GTA Production Environment

1. Upon successful completion of testing in the GTA Portal Test environment with no known level 1 or level 2 issues (based upon the attached definitions) and agreement to deploy, the application will be deployed to the portal production environment from the GTA configuration management repository using the agencies deployment instructions.
2. Once the application has been deployed in the production environment, the agency will make a test transaction through in the production environment in order to insure a successful deployment.
3. If an application error occurs in the portal production environment, the application will be returned to the development environment from which it came for problem determination and rework.